What is claimed is:

**1**. A data processing system for processing programs, the system comprising:

memory for storing data objects, the data objects being referenced by pointers; and

a short-quasi-unique-identifier (SQUID) generator which generates SQUIDs for newly allocated data objects to be stored in the memory segment, pointers to a particular data object being associated with the data object's SQUID.

**2**. The system of claim 1, further comprising:

a memory allocator which allocates a segment of the memory to a data object.

**3**. The system of claim 2, wherein if the data object is moved to a second allocated memory segment, a pointer to the second allocated memory segment is placed at the original memory segment.

**4**. The system of claim 3, wherein the data object is moved due to resizing.

**5**. The system of claim 3, wherein the data object is moved from a first memory to a second memory within a distributed system.

**6**. The system of claim 3, wherein the data object is moved due to garbage collection.

**7**. The system of claim 3, wherein the data object is moved due to data compaction.

**8**. The system of claim 1, wherein the distribution of SQUIDs over a range is uniform.

**9**. The system of claim 8, wherein SQUIDs are generated by counting.

**10**. The system of claim 8, wherein SQUIDs are generated randomly.

**11**. The system of claim 8, wherein SQUIDs are generated by hashing.

**12**. The system of claim 1, further comprising:

a comparator which compares SQUIDs associated with two different pointers.

**13**. The system of claim 12, further comprising:

an instruction reordering mechanism which reorders instructions and which is responsive to the comparator.

**14**. The system of claim 12, the comparator determining that the two pointers do not reference the same data object if the SQUIDs are different.

**15**. The system of claim 14, the comparator further determining that the two pointers reference the same data object if the SQUIDs are identical and address fields of the two pointers are identical.

**16**. The system of claim 15, each pointer address field comprising:

a base address; and

an offset,

the comparator further determining the two pointers do not reference identical locations within a referenced data object if the pointers' offsets are not identical.

**17**. The system of claim 12, wherein a pointer is associated with a migration indicator field which indicates a number of migrations of the referenced data object by the time said pointer is created, the comparator determining that said two pointers do not reference the same data object if their associated migration indicators indicate identical numbers of migrations and their corresponding addresses are different.

**18**. The system of claim 17, wherein the migration indicator comprises one bit.

**19**. The system of claim 1, wherein at least one pointer is a guarded pointer.

**20**. The system of claim 1, wherein the SQUID is implemented by hardware.

**21**. The system of claim 1, wherein the SQUID is implemented by software.

**22**. The system of claim 1, wherein a pointer contains its associated SQUID.

**23**. The system of claim 1, further comprising a SQUID cache for storing SQUIDS of recently-used pointers.

**24**. A data processing system for processing programs, the system comprising:

memory for storing data objects;

pointers to data objects stored in the memory;

migration indicators associated with pointers, a migration indicator indicating a number of migrations of a data object referenced by an associated pointer prior to said pointer being created; and

a comparator which determines that said two pointers do not reference the same data object if their associated migration indicators indicate identical numbers of migrations and their corresponding addresses are different.

**25**. The system of claim 22, wherein the migration indicator comprises one bit.

**26**. The system of claim 22, wherein the migration indicator comprises multiple bits.

**27**. The system of claim 22, wherein the migration indicator is implemented by hardware.

**28**. The system of claim 22, wherein the migration indicator is implemented by software.

**29**. A method for processing programs, the system comprising:

storing data objects in a memory, the data objects being referenced by pointers; and

generating a short-quasi-unique-identifier (SQUID) and assigning the SQUID to a data object stored in the memory segment, pointers to the data object being associated with the data object's assigned SQUID.

**30**. The method of claim 27, further comprising:

allocating a segment of the memory to the data object.

**31**. The method of claim 28, wherein if the data object is moved to a second allocated memory segment, a pointer to the second allocated memory segment is placed at the original memory segment.

**32**. The method of claim 29, wherein the data object is moved due to resizing.

**33**. The method of claim 29, wherein the data object is moved from a first memory to a second memory within a distributed system.

**34**. The method of claim 29, wherein the data object is moved due to garbage collection.

**35**. The method of claim 29, wherein the data object is moved due to data compaction.

**36**. The method of claim 27, wherein the distribution of SQUIDs over a range is uniform.